International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

A New Approach for Real Time Evidence Collection from Linux Environment

Neethu P Nair[1], Maniveena C[2]

[1]Cochin University of Science and Technology

[2] College of Engineering Kallooppara, India

Abstract: ***Evidence collection from computers is an important step in the process of digital investigations. An event could correspond to a system log entry where the operating system has recorded that a particular user or application performs a certain action. Depending on the configuration of the system the logs may omit some types of forensically interesting events and include various forensically uninteresting events. So there is an increased need of a system that will collect evidences related to computer activities. Through this paper a real time computer forensics system that records computer activity for forensic investigation on a Linux based computer system is aimed. This will help investigators who look for evidences in these operating systems. This method is different from the traditional post-mortem method of examining data since activities are being recorded as they are happening.***

Keywords: inode, inotify, post-mortem analysis, syslog, monitoring

1. Introduction

We all know that computer systems log each and every action taking place in it which is known as events. The large volume of events recorded in a log could be massive, sometimes concealing the interesting events, that makes searching for them a big problem. This problem is termed as quantity or verbosity problem. The Verbosity Problem in Digital Forensics is that the quantity of data to examine can be very large. It is inefficient to scrutinize every single piece of it. Data reduction techniques are used to resolve this, i.e., by removing known data or taking only interested data. Relying on system logs for evidence collection is an unsound manner as system logs tend to give inaccurate results. Due to the difficulty of forensic investigations, the design of new methods and tools for speeding up and automating tasks required by digital forensic processes has become a challenging task. In particular, the collection of digital evidence is an intricate work that needs special care.

Coming to Linux we know that it is a member of Unix family. Linux was built from the ground up with safety and hardware compatibility in mind, and is currently one of the most popular Unix-based operating systems[1].Its lack of popularity amongst normal users and popularity amongst programmers and intruders is one reason Linux has not been a profitable target. In the past few years, popularity has increased for UNIX based operating systems such as Mac OS, Android, and Linux. As a result, these platforms have become a more profitable prey for attackers. Ubuntu is an open source Debian based Linux operating system and distribution for personal computers, smart phones and network servers. Another thing to notice is that Linux treats its devices as files. The directory where these "files" are maintained is "/dev". The ext2, ext3 and ext4 are all file systems created for many distributions of the Linux operating system. The metadata for each file and directory are preserved in a data structure called an inode, which has a fixed size and is located in an inode table [2].In Linux we have to gather information from scattered sources [3]. Linux system does not have a registry, and file system is also different from windows operating systems. Another speciality is that, in Linux files/data are stored mostly in plain text. So it is good for string searching and interpreting data. Linux systems have great potential for both being very secure and being exploited [4].

The starting point for any computer investigation needs to be an analysis of log files in the system. Syslog is designed to provide the ability to report system events.Log files on Linux systems are interesting things. They are highly configurable, efficient and detailed. Linux logs are in clear text, so you will not need to use any third-party software or utilities to perform effective searches [5]. The data passed by the syslogd daemon to a remote syslogd process is sent in clear text [6]. It is observable using a network analysis tool such as tcpdump. This allows an invader to view the events being reported by routers or switches and by tools such as Snort5. The invader can then use this information to determine which systems they need to attack. So the problem in syslog is that it lacks confidentiality.

So there is an increased need of a system that will collect evidences related to computer activities. A real time computer forensics system that records certain types of computer activity is helpful.This paper proposes a real time computer forensics system that records computer activities or events on a Linux based computer system. This is very much useful for investigators who look for evidences in these operating systems.

2. Proposed System

The proposed system is different from other existing logging systems as it catches real-time activities carried out by the users such as adding, deleting, or modifying a file and connecting a removable device. These things are logged in real time as the user actually performs the actions. Instead of relying on system logs, this is a framework for logging events specifically with forensic purposes in mind. It consists of

certain modules for capturing and logging different types of computer activity in a normalized format to more easily facilitate advanced analysis and data reduction techniques. An application that runs on each system and captures various user and software activities on that system while logging them to a database for future analysis.

This will enable forensic investigators to collect computer activity data samples from Linux operating system. These details can then be investigated for research. The traditional computer forensics investigative methodology is both time taking and costly in terms of human resources and laboratory software and hardware [7]. The acquisition of data from storage media post-mortem poses critical efficiency problems for investigators, as the size of computer storage media increases corresponding with an expansion of computer related activities. So the proposed way seems to be helpful.

Mainly these types of data are collected by the system. These are organised into modules and they are,

- File system event watcher: It records the activity taking place in the file system.
- Removable device discovery: It records whether any external devices are plugged into the system.

In the proposed system each module carries out a specific function. Each module collects events and stores it into a database. The output is saved so as to enable further analysis. The database contains the details about the file system events happened in a system, and USB connection details. These details stored in the database can be retrieved by an investigator by creating an interface.

2.1 File system event watcher

File system is responsible for storing information on disk and retrieving and updating this information. In Linux all the things are treated as a file. When dealing with Linux machines, Ext based file systems will be found frequently.Ext3 is an extension of ext2 that added journaling to the mix to relieve us of those horribly long file system checks when something has gone awry. Ext4 is a deeper enhancement over Ext3 than Ext3 was over Ext2.

The need to scan a given file system for changes is a fairly common one, and there are a number of common tasks which need this, including:

- Notifying applications of alterations in configuration files.
- Tracking modifications in critical system files.

A common approach to doing this sort of change notification is file polling, however this appears to be ineffective for all but the most frequently-changed files and can miss some types of changes. Data integrity systems like Tripwire track file changes based on a fixed time schedule, but the time-scheduled method doesn't work if you want to be notified every time it changes in real-time just as an event takes place. A framework which fulfills that requirement is Inotify. Here also we are using this functionality.

Inotify is a file change notification system in the Linux kernel, accessible since version 2.6.13. [8]. It is an efficient way to trace actions in the file system on Linux in real-time. The inotify API provides a method for monitoring file system events. It can be used to monitor individual files or to monitor directories. When a directory is noticed inotify will give events for the directory itself, and files inside the directory. It replaces an earlier facility called dnotify which had similar goals. One major use is in desktop search utilities where its functionality allows re indexing of changed file system for changes every few minutes, which would be very inefficient. The kernel directly told that a file has changed. But inotify does not support repeatedly watching directories, meaning that a different inotify watch must be created for each sub directory.

2.1.1 Inotify Execution flow

- Creating inotify monitoring (watch) list. Add the directories/files to the inotify monitoring (watch) list that you want to watch. We can change the monitoring list as and when needed.
- Request Inotify to report particular event changes to the monitoring list of files and directories. For example, we can request inotify to perform checking and report events such as ON ACCESS, ON OPEN, ON WRITING, ON CLOSE etc.,

2.1.2 Used functions

- Creating the inotify instance by inotify_init().The more recent inotify_init1() is like inotify_init() but it will provide some extra service [9].
- Adding all the directories to be monitored to the inotify list using inotify_add_watch() function. This function manipulates the watch list associated with an inotify instance. Each item ("watch") in the watch list specifies the path name of a file or directory with some set of events that the kernel should watch or monitor for the file referred to by that pathname.
- To determine the changes or events happened, perform read() on the inotify instance. This read will get blocked until the change event occurs. Read returns list of events happened on the monitored directories. On the basis of the return value of read(), we will know precisely what kind of changes occurred.
- In case of separating or removing the watch on directories/files, call inotify_rm_watch().

2.1.3 Events

- IN_ACCESS – File was accessed[9]
- IN_ATTRIB – Metadata (data that describes other data) changed (Permissions, timestamps, extended attributes, etc.)
- IN_CLOSE_WRITE – It means that the file opened for writing was closed
- IN_CLOSE_NOWRITE – This event refers that, file not opened for writing was closed
- IN_CREATE – File/directory was created in watched directory
- IN_DELETE – File/directory was deleted from watched directory

- IN_DELETE_SELF – Watched file/directory was itself deleted. This event also occurs if an object is moved to another filesystem ,since in effect copies the file to the other filesystem and then deletes it from the original filesystem
- IN_MODIFY – It means that the file was modified
- IN_MOVE_SELF – Watched file/directory was itself moved
- IN_MOVED_FROM – A file was moved out of watched directory
- IN_MOVED_TO – File moved into watched directory
- IN_OPEN –File was opened

3. Conclusion

A computer-related investigation often produces a particularly large volume of evidence [11]. Managing all this data and using it effectively throughout the span of an investigation presents special problems. The proposed system will log certain activities taking place in a computer system that may help forensic examiners. The proposed system is different than other existing logging systems as it captures real-time activities taken by the users such as adding, deleting, or modifying a file and connecting a USB device.

4. Acknowledgement

We would like to thank, first and foremost, Almighty God, without his support this work would not have been possible. We would also like to thank all the faculty members of College of Engineering Kallooppara for their immense support

References

[1] Ubuntu manual team, Getting started with Ubuntu 14.04
[2] https://en.wikipedia.org/wiki/Inode
[3] Linux Forensics (for Non -Linux Folks) Hal Pomeranz Deer Run Associates
[4] Network Security Bible Second Edition, Eric Cole
[5] Unix and Linux forensic analysis DVD toolkit, Chris Pogue, Cory Altheide, Todd Haverkos
[6] Kenneth E. Nawyn, " A Security Analysis of System Event Logging with Syslog", SANS Institute, 2003.
[7] Shadi Al Awawdeh, Ibrahim Baggili, Andrew Marrington, FarkhundIqbal, "CAT Record (Computer Activity Timeline Record): A unified agent based approach for real time computer forensic evidence collection", IEEE 2013.
[8] http://people.clarkson.edu/
[9] http://www.thegeekstuff.com/
[10] http://askubuntu.com/
[11] https://www.fbi.gov/

Author Profile

Neethu P Nair obtained B.Tech in Computer Science and Engineering from Mount Zion College of Engineering, Kadammanitta in 2013. She is now pursuing her master degree in Computer Science with specialization in Cyber Forensics and Information Security at College of Engineering, Kallooppara under Cochin University of Science and Technology.

Maniveena C obtained B.Tech and M.Tech in Computer Science and Engineering from Anna University. She is currently working as Assistant Professor in College of Engineering, Kallooppara.